

## Interactive multi-objective particle swarm optimisation using decision space interaction

Hettenhausen, Jan; Lewis, Andrew; Randall, Marcus; Kipouros, Timoleon

*Published in:*  
2013 IEEE Congress on Evolutionary Computation, CEC 2013

*DOI:*  
[10.1109/CEC.2013.6557988](https://doi.org/10.1109/CEC.2013.6557988)

*Licence:*  
Other

[Link to output in Bond University research repository.](#)

*Recommended citation(APA):*  
Hettenhausen, J., Lewis, A., Randall, M., & Kipouros, T. (2013). Interactive multi-objective particle swarm optimisation using decision space interaction. In *2013 IEEE Congress on Evolutionary Computation, CEC 2013* (pp. 3411-3418). [6557988] Wiley-IEEE Press. <https://doi.org/10.1109/CEC.2013.6557988>

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

For more information, or if you believe that this document breaches copyright, please contact the Bond University research repository coordinator.

# Interactive Multi-Objective Particle Swarm Optimisation using Decision Space Interaction

Jan Hettenhausen\*, Andrew Lewis†, Marcus Randall‡ and Timoleon Kipouros§

\*Institute for Integrated and Intelligent Systems  
Griffith University, Brisbane, Australia  
Email: j.hettenhausen@griffith.edu.au

†Institute for Integrated and Intelligent Systems  
Griffith University, Brisbane, Australia

‡Faculty of Business  
Bond University, Gold Coast, Australia

§Department of Engineering  
University of Cambridge, Cambridge, United Kingdom

**Abstract**—The most common approach to decision making in multi-objective optimisation with metaheuristics is a *posteriori* preference articulation. Increased model complexity and a gradual increase of optimisation problems with three or more objectives have revived an interest in progressively interactive decision making, where a human decision maker interacts with the algorithm at regular intervals. This paper presents an interactive approach to multi-objective particle swarm optimisation (MOPSO) using a novel technique to preference articulation based on decision space interaction and visual preference articulation. The approach is tested on a 2D aerofoil design case study and comparisons are drawn to non-interactive MOPSO.

## I. INTRODUCTION

Solving multi-objective optimisation problems ultimately requires a human decision to select the most desirable solution within the Pareto-optimal set. This can either occur *a priori*, *a posteriori* or interactively. A search through the respective literature reveals that currently the majority of metaheuristic approaches are either *a priori* or *a posteriori* compared to a very small number of interactive approaches. This observation was also made by, for example, Figuera *et al* [1], Sinha [2] and Sinha *et al* [3].

*A priori* approaches require preference articulation before the optimisation starts. Often these preferences are then aggregated, effectively turning the multi-objective optimisation problem into a single objective one. A disadvantage of this is that algorithms may then experience difficulties finding desirable solutions on, for example, some convex Pareto-fronts.

*A posteriori* approaches, on the other hand, are designed to find a good approximation to the Pareto-optimal set in regards to convergence and spread of solutions. The decision maker can then choose from a large set of available solutions. This has led to a widespread popularity of *a posteriori* approaches. Disadvantages and criticisms include high computational costs, difficulties in dealing with problems with more than a few objectives and the complexity of the necessary *a posteriori* data analysis.

More recently the idea of interactive involvement of the decision maker has gained more interest. Interactive approaches

allow the decision maker to articulate their preferences at various times during the optimisation process. This alleviates the problem of expressing preferences without knowledge of available solutions and structure of the search space. Applying guidance to the search further reduces the computational costs by ignoring regions of the search space that are of little interest. It also allows the use of optimisation as an assisted design tool, furthering the understanding of the decision maker of the design space.

The literature base on interactive multi-objective metaheuristics is still comparatively small and most methods are based on Multi-Attribute Value Theory (MAUT). Commonly in these approaches the decision maker is asked to make a pairwise comparison of non-dominated solutions to articulate their preferences. The outcome of these comparisons is then used to construct a value function in objective space which augments or replaces the dominance relationship. The majority of these approaches are currently based on Evolutionary Algorithms. Such methods were, for example, developed by Deb and Kumar [4], Deb *et al* [5], Sinha *et al* [6] and Yadav *et al* [7]. A method for discrete optimisation was presented by Phelps *et al* [8]. A MAUT-based algorithm using MOSPO was developed by Agraval *et al*. [9]. It follows the same ideas as the methods based on Evolutionary Algorithms in its use of a utility function based on pairwise comparison. A good overview of approaches based on MAUT based interactive Evolutionary Algorithms can be found in Deb *et al*. [5].

A non MAUT-based approach to interactive MOPSO was presented by Hettenhausen *et al* [10]. Their approach allows the decision maker to choose from all available solutions by providing a visual “heatmap” for representing and interacting with the available solutions. The method enables the decision maker to gain an understanding of the available set of solutions through visual analysis. They could then select one or more points as desirable guides for the swarm.

In this paper a novel approach to interactive MOPSO is presented. The idea of a visualisation-based user interface is used and combined with Parallel Coordinates visualisation [11] to gather the decision maker’s preferences which take the form of boundary constraints in parameter space. Inspiration

for this was taken from workflows in aeronautics engineering where for large scale problems multiple runs of *a posteriori* optimisation with iteratively refined boundary constraints are used. The remainder of this paper has the following structure. Relevant definitions of multi-objective optimisation and a brief overview of MOPSO will be given in Section I-A and II respectively. Section III presents the proposed user interaction, followed by an explanation of the underlying algorithm that integrates the user preferences in Section III-B. Computational experiments based on an aerofoil case study are discussed in IV. Concluding remarks and a brief outlook on future work can be found in Section V.

### A. Multi-Objective Optimisation

Without loss of generality, only minimisation problems will be considered; any objective function  $f(x)$  requiring maximisation can be transformed into one requiring minimisation using a substitute function  $h(x) = -f(x)$ . A multi-objective objective can then be formally expressed as:

$$\begin{aligned} \text{minimise } \vec{f}(\vec{x}) &= \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})\} \\ f_{k \in [1, m]} : \mathbb{R}^n &\rightarrow \mathbb{R}, \forall \vec{x} \in \mathcal{S} \subseteq \mathbb{R}^n \end{aligned} \quad (1)$$

with parameter vectors  $\vec{x} \in \mathbb{R}^n$ , the feasible search space  $\mathcal{S}$  formed by a constraint function and  $m \geq 2$  objectives. The feasible space  $\mathcal{S}$  is commonly denoted the *decision space* or *parameter space*. The image of the decision space, subject to  $\vec{f}$ , is called the *objective space*. Its elements  $\{f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})\} \in \mathbb{R}^m$  are referred to as *objective vectors*.

A partial order can be imposed on the decision space via the dominance relation. A decision vector  $\vec{x}_1$  dominates another decision vector  $\vec{x}_2$ , denoted  $\vec{x}_1 \prec \vec{x}_2$ , iff it is not worse in all objectives, i.e.  $\vec{f}(\vec{x}_1, i) \leq \vec{f}(\vec{x}_2, i), \forall i \in 1, \dots, m$ , and strictly better in at least one objective, i.e.  $\exists i \in 1, \dots, m : \vec{f}(\vec{x}_1, i) < \vec{f}(\vec{x}_2, i)$ . If only the first of these conditions is fulfilled,  $\vec{x}_1$  is said to weakly dominate  $\vec{x}_2$ . Decision vectors are called Pareto-optimal if they are globally non-dominated. The set of all non-dominated decision vectors is the Pareto-optimal set. The image of the Pareto-optimal set in objective space is the Pareto-optimal front, or Pareto-front. The dominance relation cannot further distinguish between decision vectors of the Pareto-optimal set, necessitating the incorporation of user preferences to determine the desirability of the available Pareto-optimal decision vectors.

## II. PARTICLE SWARM OPTIMISATION

Particle Swarm Optimisation (PSO) is a population based metaheuristic first developed by Kennedy and Eberhart [12]. Notable variations of PSO for multi-objective optimisation were developed by Hu and Eberhart [13], Coello Coello and Lechuga [14], Mostaghim and Teich [15] and Fieldsend and Singh [16].

All variations of dominance based MOPSO share the same core principles. A swarm of  $N$  particles is maintained by the algorithm and remains constant during the run. The swarming behaviour of the particles is governed by a velocity equation. The velocity is dependent on the weighted previous velocity and two additional points  $p_{best}$  and  $p_{gbest}$ , representing the

particle's, and the entire swarm's, knowledge of good solutions respectively. For each particle the velocity  $v$  in iteration  $t + 1$  can be stated as:

$$\vec{v}_{t+1} = w\vec{v}_t + c_1r_1(p_{gbest} - \vec{x}_t) + c_2r_2(p_{pbest} - \vec{x}_t) \quad (2)$$

where  $x_t$  is the position of the particle in iteration  $t$ ,  $w$  is the inertia weight,  $c_1$  and  $c_2$  are constant positive weights and  $r_1$  and  $r_2$  are uniform random weights.  $p_{best}$  and  $p_{gbest}$  are commonly denoted as the cognitive and social component. Algorithmically they are represented by archives of non-dominated solutions. The  $p_{pbest}$  is specific to each particle, containing only non-dominated solutions found by this particle. Depending on the implementation it can either contain one or more points. The global  $p_{gbest}$ , on the other hand, contains the non-dominated points discovered by the entire swarm. Based on the velocity  $\vec{v}_{t+1}$  the position of the particle is updated using the equation:

$$x_{t+1} = x_t + \vec{v}_{t+1} \quad (3)$$

A range of approaches exists for the global guide particle selection. The following is an overview of a few of the most common methods.

Coello Coello and Lechuga [14] proposed dividing the objective space into evenly sized hypercubes. Each hypercube is then assigned a score that is inversely proportional to the number of non-dominated particles within its bounds. Roulette wheel selection is applied to select a non-empty cube; the actual global guide is then selected randomly from the selected cube. The method aims at promoting cubes with fewer non-dominated points and with it a more even coverage of the Pareto-front.

In contrast, the Sigma method by Mostaghim and Teich [15] has a stronger focus on promoting convergence rather than even spread. Their method selects guide particles based on smallest weighted angle between a vector from the origin to a given particle and equivalent vectors for the known non-dominated points. The authors note that for good diversity a larger population is ideal.

An approach based on a tree-structure for the archive was developed by Fieldsend and Singh [16]. In their approach the archive is not constrained in any way. This is possible because of the reduced computational cost of their archiving method. Global guide particle selection is based on choosing weak dominance and proximity in objective space.

For more comprehensive overviews of guide particle selection mechanism, the reader is referred to Mostaghim and Teich [15], Engelbrecht [17] and Fieldsend and Singh [18].

## III. USER INTERACTION

In this paper a novel approach to user interaction is described. While commonly for interactive optimisation preferences are articulated in objective space, the algorithm described in this paper operates on preferences in parameter space. The inspiration for this approach lies in the design cycles often used in conjunction with *a posteriori* algorithms on large scale problems (e.g. Kipourous *et al* [19]). For suitably

large problems consecutive runs of an *a posteriori* algorithm with increasingly narrow boundary constraints are used. Manual post analysis is performed to define new border constraints for the following run.

Based on this procedure, an interactive optimisation algorithm was developed. In order to effectively employ human reasoning and domain knowledge, the approach is centred around a visual representation of the decision and objective space. Consequently, the user interface represents an integral part of the approach to interaction proposed in this paper. While in the post analysis of results from *a posteriori* algorithms visualisation is already commonplace, interactive approaches, particularly those based on MAUT, do not generally incorporate visualisation as user interface. This algorithm takes conceptual inspiration from Hettenhausen *et al* [10] where the user interaction is driven by visualisation.

In the method suggested in this paper, the decision maker is asked to review the current progress at regular intervals using this user interface. The user is then given the opportunity to set, delete or adjust boundary constraints in decision space. These boundary constraints are then used by the algorithm to bias the search accordingly for a set number of iterations until the next interaction. The following section outlines the user interactions as well as the suggested mode of interaction. Section III-B then discusses the integration with the algorithm.

#### A. User Interface

The user interface comprises three different visualisations: a parallel coordinates plot, a scatter plot and a visualisation of a selected solution. Figures 1 and 2 show the user interface at different stages of the optimisation process. The primary form of visualisation is done using the parallel coordinates plot. Parallel coordinates represent an elegant solution to visualise high dimensional datasets. Applied to multi-objective optimisation problems, each dimension of the parameter space and the objective space is commonly represented by a vertical axis with ranges for each axis being independent of each other. Each solution can then be represented by a line crossing each axis at the appropriate value. Structurally, Parallel Coordinates bear some resemblance to radar charts.

For each axis the user can select a range, in this case represented by a grey bar, that determines which solutions remain highlighted and which are greyed out. Applying such constraints to a number of axes allows one to visually analyse correlations between parameter space and objective space. For example, a decision maker interested in a certain combination of lift and drag can select appropriate ranges for those two objectives as shown in Figure 1. This leaves all those solutions, and parameters, highlighted whose objectives lie within the selected ranges in objective space. Comparing highlighted and unhighlighted points can often give insight into which parameters strongly correlate with desirable solutions and which parameters have a lesser influence. It can also suggest in what direction the algorithm could be guided to further improve the approximate Pareto-front, based on the decision maker's preferences. In addition to selecting ranges in objective space, the decision maker can also select ranges in parameter space to limit the view as seen in Figure 2. In the method presented, this is also the way of ultimately articulating preferences; ranges

set in parameter space can, once the decision maker is satisfied with them, be submitted as new boundary constraints to the algorithm. Their incorporation in the search will be outlined in the following Section.

The example given in Figures 1 and 2 outlines this procedure. The decision maker selects a range in objective space they consider desirable. In Figure 1 values with low values for the second objective  $o_1$  are chosen. In this case this selection is sufficient, but in other cases the selection could be further limited by selecting another range for the first objective. By now comparing highlighted versus greyed out solutions, it becomes obvious that, for example higher values for parameter six ( $p_6$ ) have a high correlation with solutions in the desired range in objective space. After selecting this range on  $p_6$  and removing the selection on  $o_1$  it can be visually verified that this selection provided a useful selection for the next few iterations (Figure 2). In other cases this could be slightly less clear and more than one parameter needs to be constrained to achieve the desired selection. When the decision maker suspects that improvements could be made by further exploration in a certain direction, they may select the range accordingly, to which the swarm will react accordingly. The mechanism for this will be discussed in Section III-B.

To improve the ability of the decision maker to better explore available solutions, they can augment their understanding of the nature of each solution by individually selecting it. Points can be selected either in the parallel coordinates plot or in the scatter plot (Figure 1 and 2). The corresponding point is then highlighted in both representations. When combining a parallel coordinates plot and a scatter plot, any two dimensions can be selected for the scatter plot.

In addition, a visualisation of the actual design corresponding to that point, in this case the aerofoil shape, is loaded in the third representation. While not necessarily possible for all optimisation problems, this representation allows the decision maker to analyse relevant points in a more tangible way. It may further allow the decision maker to identify designs that may be infeasible in practice but evaluate correctly in the simulation. Such a case was, for example, presented by Kamalian *et al.* [20] in a Microelectric Mechanical Systems application.

#### B. Integration with the Algorithm

The underlying algorithm used for this study was implemented as described in Section II. The specific algorithm chosen is that of Coello Coello and Lechuga [14], as it is one of the most widely used variants and one of the most generally applicable ones with usually little dependence on algorithm parameters. Each particle was given an archive size of one. Guide particle selection was performed using hypercubes with a  $10 \times 10$  grid.

Initial experiments focussed primarily on restricting the global best guide particle selection to particles in the global archive that were not in violation of the user specified boundary constraints. To balance exploration and focus of the swarm, the mechanisms for personal best guide selection were left unchanged. This approach proved effective in guiding the swarm towards regions desirable to the decision maker but proved to be slow promoting exploration of regions of interest.

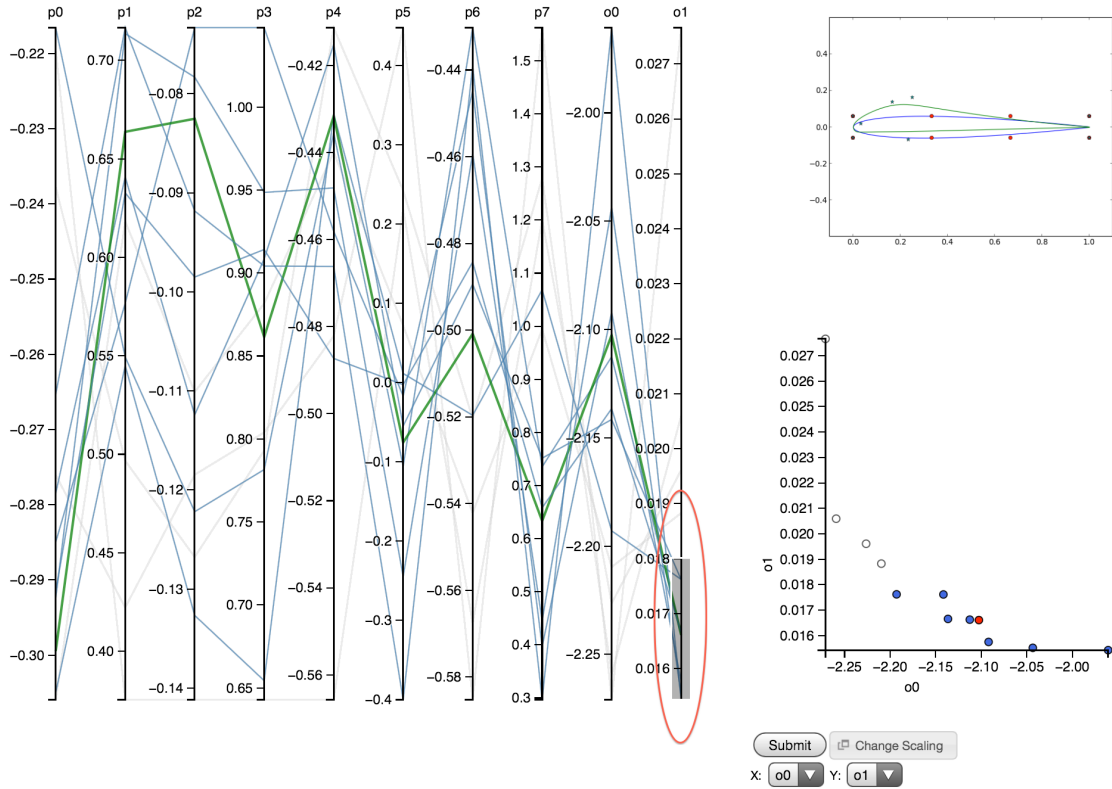


Fig. 1. User interface for interactive MOPSO. The screenshot illustrates the first step of the decision process where a range in objective space is selected to parameter ranges that correlate with desirable solutions (highlighted). In this figure the scatterplot in the lower right corner shows  $o1$  vs  $o2$ . The user can further select any of the available solutions in either the scatter plot or the parallel coordinates plot. The corresponding point will then be highlighted in both plots and the plot of the design displayed in the upper right hand corner.

Particularly when only few points close by each other are available in a selected interval, the ability of the swarm to diversify from this range could be degraded.

As a way to promote diversity and exploration in selected intervals, the concept of “virtual guide particles” was developed. Similar to the way hypercube guide particle selection aims at promoting less dense regions of the approximate Pareto-front, virtual guide particles aim at promoting regions considered desirable by the decision maker. They are added to complement the current guide particle selection mechanism rather than replacing it and occur with a set probability.

The construction of virtual guide particles follows the following schema. For each dimension in decision space a list of values that satisfy this constraint is created from known solutions in the archive.

- If one or no such points are found, a random one is generated using a Gaussian distribution around the centroid of the upper and lower boundary with a standard deviation of about 10% of the range.
- If more than one valid value is found and no additional boundary constraint is set by the decision maker, one value is chosen randomly and a small turbulence value is applied to it.
- If more than one valid value is found and a user specified boundary constraint is defined the parameter value is chosen based on coverage of the range:

- If less than 80% of the selected range are covered by known points, a point is chosen at random within the largest gap. The procedure is again based on a Gaussian distribution with identical parameters as above. The assumption is, that the decision maker would like to further explore the selected range.
- If the coverage is 80% or more, it is assumed that the decision maker simply wants to restrict the search to this range. In this case an existing point within the range is chosen and a small turbulence applied.

Other than normal guide particles, virtual guide particles are not known to be good or even valid points. They, however, are a representation of the decision maker’s assumption about potentially good solutions. As parameters are selected independent of each other, rather than being a linear combination of known points, the construction mechanism is somewhat related to the cross-over operator in Genetic Algorithms. A side effect of it is, that combinations can be produced that are more difficult to achieve by a linear combination of parameters.

#### IV. COMPUTATIONAL EXPERIMENTS

The method was evaluated using a case study based on a special configuration of the 2D aerofoil project. The following sections will first introduce the case study, followed by an overview of the configurations used for algorithms evaluated

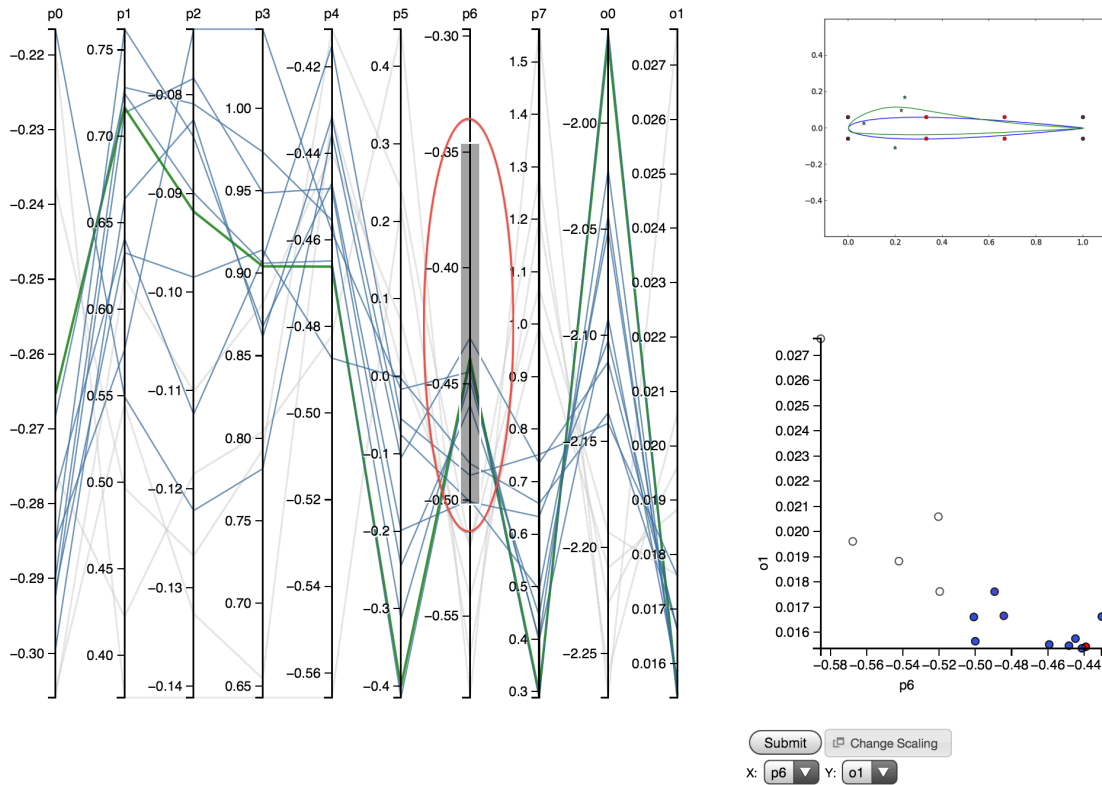


Fig. 2. User interface for interactive MOPSO. The screenshot illustrates the second step of the decision process where a new boundary constraint is defined on parameter  $p_6$  (highlighted). The scatter plot in this figure shows  $p_6$  vs  $o_1$  to illustrate the effect of the selection. In the above case a reasonably good separation could be achieved with this selection. If a more narrow range of points is desired, the most obvious parameters one might choose to constrain would be  $p_0$ ,  $p_4$  and possibly  $p_5$ , depending on the desired outcome.

and a discussion of the results. The principal aim of the evaluation was to demonstrate the ability of interactive MOPSO to effectively follow the guidance of a human decision maker and achieve superior performance in a region of interest, compared to its non-interactive counterpart.

#### A. Case Study

As a case study, a special case of the 2D aerofoil benchmark problem was used. The design of the case study is closely based on the case study used by Kipouros *et al.* [21] and it closely resembles challenges encountered in a real world compressor blade design problem that is still under active investigation. As the basis the NACA0012 aerofoil [22] was used, an aerofoil with symmetric top and bottom surfaces (no camber) and a maximum thickness of 12% of the width of the aerofoil. Additional thickness constraints are applied at 25% and 50% of the chord.

The particular configuration used here features two objectives, based on the lift and drag coefficients, and is closely inspired by the one used by [21]. Due to the symmetry of the NACA0012 aerofoil, it has relatively poor performance characteristics compared to an optimised, non-symmetric aerofoil.

As parametrisation for the problem, the Free Form Deformation (FFD) method, first published by Sederberg and Parry [23], was used. FFD is based on trivariate Bernstein polynomials and allows definition of a deformation, i.e. a  $R^2 \rightarrow R^2$  mapping of all points in a specified rectangular

regions, using a set of control points. For this study eight control points on a four by two grid were used. The algorithms in this study were given the ability to manipulate four of these control points in  $x$ - and  $y$ -dimension, giving the problem a total of eight parameters. These control points are denoted as  $L_1, L_2, L_3$  and  $L_4$  respectively. The remaining four control points were assumed fixed to ensure a constant angle of attack and chord length. An illustration of the parametrisation can be found in Figure 3. The geometry shown is the datum geometry. Formally, the parameter vector can be expressed as:

$$\vec{x} = [L_1^x L_1^y L_2^x L_2^y L_3^x L_3^y L_4^x L_4^y] \quad (4)$$

The objectives are defined based on the lift and drag coefficients of a candidate solution  $C$ . Using the additive inverse of the lift objective, the problem can be treated as a minimisation problem. Formally the objectives can be expressed as:

$$f_1(\vec{x}) = -C_L \quad (5)$$

$$f_2(\vec{x}) = C_D \quad (6)$$

The objective function values for candidate aerofoils were obtained using the Open Source Software Xfoil [24] in version 6.97. Solutions for which non-convergence occurred at any step were considered invalid, independent of whether Xfoil yielded a valid output for them.

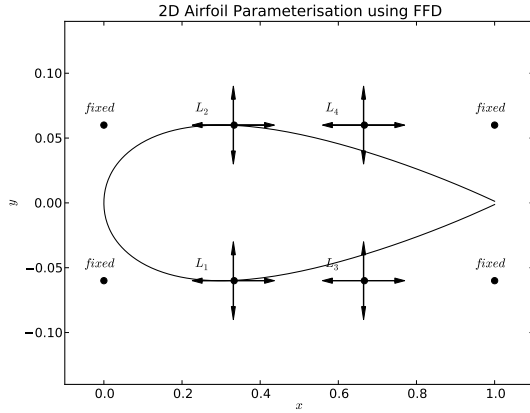


Fig. 3. Parameterisation of the 2D aerofoil problem using FFD. Points  $L_1, L_2, L_3$  and  $L_4$  are manipulable by the algorithm.



Fig. 4. Illustration of valid (black) and invalid (white) points found by particles in a representative run with non-interactive MOPSO.

The resulting optimisation problem features a substantial number of invalid points within the search space. They can, amongst other reasons, be caused by physically impossible shapes produced by the parametrisation, violation of thickness constraints or non-convergence of any aspect of the simulation. To the algorithm this is transparent and the viability of a point is only available after simulation. Figure 4 illustrates valid and invalid points encountered during a representative run of non-interactive MOPSO. Similar tests were conducted using the NSGA-2 algorithm [25], yielding similar results but generally falling behind the MOPSO runs in terms of convergence. The seemingly random distribution of valid and invalid points suggests that the search space does not feature discernible regions with either valid or invalid points but rather features valid and invalid points side by side throughout the entire search space. The graphic further illustrates that the algorithm encounters increasingly fewer valid points the closer it approaches the true Pareto front.

### B. Experiment Configuration

In all runs the algorithm was allowed 6000 objective function evaluations. These were divided into 100 iterations with a population of 60. Interactive runs comprised an initial phase of 25 non-interactive iterations after which the decision

maker was queried every 5 iterations. This interval was chosen as it allows the decision maker to quickly react to a changed solution set.

All MOPSO variations used a momentum of  $w = 0.4$ , constant weights of either  $c_1 = c_2 = 2.0$  or  $c_1 = 2.0, c_2 = 1.0$  and random weights  $r_1$  and  $r_2$  in the range  $[0, 1]$ . As a uniform random population generally does not yield a usable number of valid solutions (see Section IV-A), the initial populations were generated using a Gaussian distribution around the datum geometry with a mean of  $\mu = 0.0$  and a standard deviation of  $\sigma = 0.2$ . Invalid points were assigned a value of 100 for both objectives, placing them far outside the range of valid points.

For the interactive runs the decision maker was instructed to aim for solutions with high lift but with drag values of at most  $f_2(\vec{x}) = 0.018$ . The region is based on the optimal region used by Kipourous *et al.* [21]. However, due to a different configuration used in their study, a comparison between results could not be made.

### C. Experimental Results

This Section will discuss the results of the computational experiments conducted and draw comparisons to two configurations of a near identical non-interactive MOPSO. Each algorithm was run ten times of which the best, median and worst run were determined. Figure 5 shows the outcomes of these runs.

As discussed in the previous section, the decision maker in the interactive MOPSO runs was instructed to aim for solutions with  $f_2(\vec{x}) \leq 0.0180$  (drag objective). To illustrate the ability of interactive MOPSO to focus on this area, the percentage of the archive satisfying  $f_1(\vec{x}) \leq -2.1000$  and  $f_2(\vec{x}) \leq 0.0180$  was evaluated based on median values of the ten runs for each algorithm. Table I and Figure IV-C show the results of this analysis.

The results show that interactive MOPSO was able to archive significantly better results in the target region  $f_2(\vec{x}) \leq 0.0180$ . Since the focus of the decision maker was to maximise lift in this range, this region is generally converged substantially further than the rest of the front generated by interactive MOPSO. This behaviour is particularly prominent in Figure 5(a) on the example of the best runs observed. The improved convergence of interactive MOPSO in the region of interest becomes even more apparent when taking the median and worst run into consideration. Interactive MOPSO consistently outperformed the other variants in terms of convergence in the region of interest while finding fewer points outside this area.

To some extent coverage was traded off for improved convergence leading to less dense fronts. However, this can in part be attributed to the difficulty of finding valid solutions the closer progressed towards the presumed true Pareto-front the swarm is (see Figure 4). Another observation is the smaller variation in convergence between best and worst run in interactive MOPSO. The largest fluctuation could be observed in non-interactive MOPSO with  $c_1 = c_2 = 2.0$ . A possible explanation for this lies in the less explorative behaviour of swarms with a velocity bias towards the global best. The general consensus appears to be that the  $c_1$  and  $c_2$  weights represent the trade-off between exploration and exploitation.



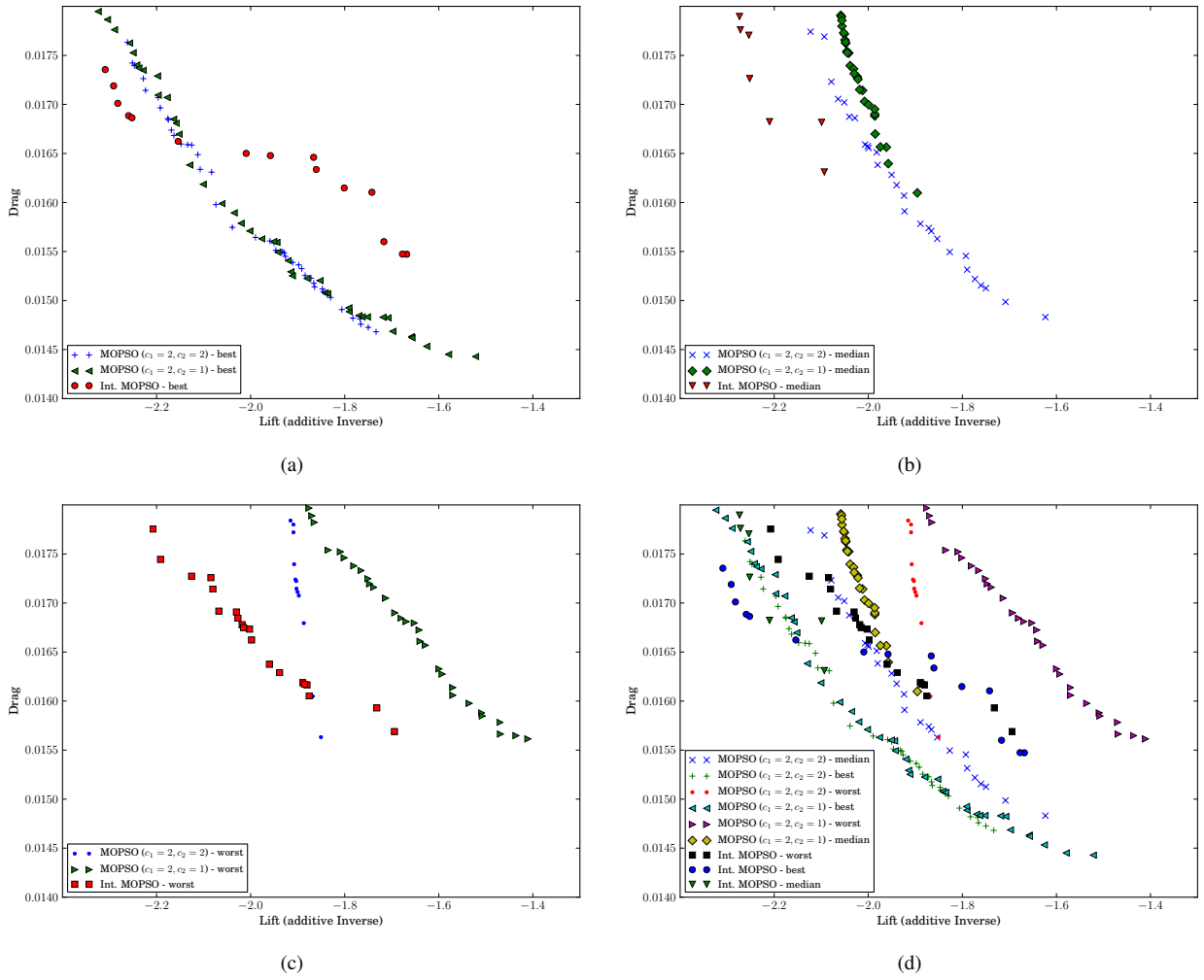


Fig. 5. Analysis of ten runs with Interactive MOPSO and MOPSO in two configurations: (a) best; (b) median; (c) worst; and, (d) all runs.

For equal weights this is assumed to be a balanced swarm. The more exploitative behaviour of the second non-interactive MOPSO likely indicated a higher dependency on the quality of the initial population. This was successfully compensated by adding user interaction.

The time requirements for each interaction were relatively short and the decision maker strictly followed the pattern described in Section III-A. While it is anticipated that in real world applications the domain knowledge of the decision maker will help navigating towards promising areas, the structure of this problem required the decision maker to carefully navigate the fitness landscape by analysing parameters in each interaction. To rule out that any potential biases based on *a priori* knowledge had an impact on the experiments, an additional series of experiments was conducted where ranges around known good values were set immediately at iteration 25. These runs performed significantly worse than the three variants analysed above.

In summary, the interactive MOPSO could achieve superior convergence over non-interactive MOPSO in the focus region and showed greater robustness to the fluctuations in the quality of the initial population.

TABLE I. MEDIAN PERCENTAGE OF ARCHIVED POINTS SATISFYING  $f_1(\vec{x}) < -2.1$  AND  $f_2(\vec{x}) < 0.180$ .

| Dataset                           | $t = 25$ | $t = 50$ | $t = 75$ | $t = 100$ |
|-----------------------------------|----------|----------|----------|-----------|
| MOPSO ( $c_1 = 2, c_2 = 2$ )      | 0.00%    | 10.59%   | 26.18%   | 26.18%    |
| MOPSO ( $c_1 = 2, c_2 = 1$ )      | 0.00%    | 14.71%   | 22.97%   | 38.57%    |
| Int. MOPSO ( $c_1 = 2, c_2 = 1$ ) | 0.00%    | 14.55%   | 28.85%   | 50.00%    |

## V. CONCLUSION

A method for interactive Multi-Objective Particle Swarm Optimisation was introduced and evaluated with a case study. The proposed method allows a decision maker to guide the search by visually analysing available solutions and apply boundary constraints on parameters where deemed necessary. The approach takes inspiration from iteratively applied *a posteriori* optimisation applied on large scale problems in engineering and possibly other disciplines.

To improve diversity of the swarm within the constraint parameter ranges, the concept of virtual guide particles was introduced. Virtual guide particles are constructed based on the constraints set by the decision maker. In each iteration a small number of virtual guides are used instead of guides the swarm would usually choose.

A comparison was drawn between interactive MOPSO



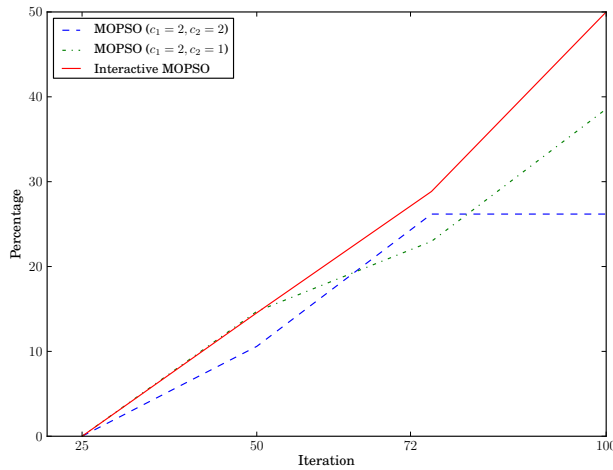


Fig. 6. Median percentage of archived points satisfying  $f_1(\vec{x}) < -2.1$  and  $f_2(\vec{x}) < 0.180$ .

and two otherwise identical configurations of non-interactive MOPSO. The interactive algorithm was consistently able to achieve better convergence in the region the decision maker targeted. This improved convergence was traded off for inferior convergence outside the focus region. Interactive MOPSO could also be demonstrated to have a more consistent performance in terms of convergence than the non-interactive variants tested.

In the future, the impact of the percentage of virtual guide particles versus common guide particles will be investigated. Different approaches to global guide selection, such as the Sigma method, will be subject to evaluation in the context of interactive MOPSO. Lastly, an often proclaimed advantage of interactive optimisation techniques is their ability to be less susceptible to the curse of dimensionality in objective space. An investigation on the applicability of the method presented on problems with three and four objectives will be conducted.

## REFERENCES

- [1] J. R. Figueira, S. Greco, V. Mousseau, and R. Słowiński, "Multi-objective optimization," in *Interactive and Evolutionary Approaches*, J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Eds. Berlin, Heidelberg: Springer-Verlag, 2008, ch. Interactive Multiobjective Optimization Using a Set of Additive Value Functions, pp. 97–119.
- [2] A. Sinha, "Progressively interactive evolutionary multiobjective optimization," Ph.D. dissertation, Aalto University, School of Economics, Department of Business Technology, 2011.
- [3] A. Sinha, P. Korhonen, J. Wallenius, and K. Deb, "An improved progressively interactive evolutionary multi-objective optimization algorithm with a fixed budget of decision maker calls," Aalto University, School of Economics, Department of Information and Service Economy, Tech. Rep., 2011.
- [4] K. Deb and A. Kumar, "Interactive evolutionary multi-objective optimization and decision-making using reference direction method," in *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2007, pp. 781–788.
- [5] K. Deb, A. Sinha, P. J. Korhonen, and J. Wallenius, "An interactive evolutionary multi-objective optimization method based on progressively approximated value functions," *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 5, pp. 723–739, 2010.
- [6] A. Sinha, K. Deb, P. Korhonen, and J. Wallenius, "Progressively interactive evolutionary multi-objective optimization method using generalized polynomial value functions," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.
- [7] S. R. Yadav, Y. Dashora, R. Shankar, F. T. S. Chan, and M. K. Tiwari, "An interactive particle swarm optimisation for selecting a product family and designing its supply chain," *Int. J. Comput. Appl. Technol.*, vol. 31, no. 3/4, pp. 168–186, 2008.
- [8] S. P. Phelps and M. Köksalan, "An interactive evolutionary metaheuristic for multiobjective combinatorial optimization," *Management Science*, vol. 49, no. 12, pp. 1726–1738, 2003. [Online]. Available: <http://www.jstor.org/stable/4133980>
- [9] S. Agrawal, Y. Dashora, M. Tiwari, and Y.-J. Son, "Interactive particle swarm: A pareto-adaptive metaheuristic to multiobjective optimization," *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, vol. 38, no. 2, pp. 258–277, March 2008.
- [10] J. Hettenhausen, A. Lewis, and S. Mostaghim, "Interactive multi-objective particle swarm optimization with heatmap-visualization-based user interface," *Engineering Optimization*, vol. 42, no. 2, pp. 119–139, February 2010.
- [11] A. Inselberg, "The plane with parallel coordinates," *The Visual Computer*, vol. 1, no. 2, pp. 69–91, 1985.
- [12] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, 1995.
- [13] X. Hu and R. Eberhart, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 2. IEEE, 2002, pp. 1677–1681.
- [14] C. Coello and M. Lechuga, "MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 2. IEEE Computer Society, 2002.
- [15] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization," in *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. IEEE Computer Society, 2003, pp. 26–33. [Online]. Available: [citeseer.ist.psu.edu/mostaghim03strategies.html](http://citeseer.ist.psu.edu/mostaghim03strategies.html)
- [16] J. Fieldsend and S. Singh, "A multi-objective algorithm based upon particle swarm optimisation," *Proceedings of The UK Workshop on Computational Intelligence*, pp. 34–44, 2002.
- [17] A. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, Ltd, 2005.
- [18] J. E. Fieldsend and S. Singh, "On the selection of gbest, lbest and pbest individuals, the use of turbulence and the impact of inertia in multi-objective pso," 2002.
- [19] T. Kipouros, M. Mleczko, and A. Savill, "Use of parallel coordinates for post-analyses of multi-objective aerodynamic design optimisation in turbomachinery," *AIAA Paper*, vol. 2138, 2008.
- [20] R. Kamalian, H. Takagi, and A. M. Agogino, "Optimized Design of MEMS by Evolutionary Multi-objective Optimization with Interactive Evolutionary Computation," in *GECCO-2004*, 2004, pp. 1030–1041.
- [21] T. Kipouros, T. Peachey, D. Abramson, and A. Savill, "Enhancing and developing the practical optimisation capabilities and intelligence of automatic design software," AIAA-2012-1677, Tech. Rep., 2012.
- [22] E. Jacobs, K. Ward, and R. Pinkerton, *The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel*. US Government Printing Office, 1933.
- [23] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," *SIGGRAPH Comput. Graph.*, vol. 20, no. 4, pp. 151–160, Aug. 1986. [Online]. Available: <http://doi.acm.org/10.1145/15886.15903>
- [24] M. Drela, "Xfoil: An analysis and design system for low reynolds number airfoils," 1989.
- [25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.